New Jersey's Science & Technology University

THE EDGE IN KNOWLEDGE

# CS 280 Programming Language Concepts

<map>

New Jersey's Science & Technology University

COLLEGE OF COMPUTING SCIENCES

## <map>

- #include <map>
- A dictionary that associates a value of one type with a value of another type
- Declaration tells the types of the items being connected
- First value is the key, second value is what the key is mapped to
- The keys in the map are unique

## <map>

- Example:

```
map<string,string> myMap;
```

- Creates an object that maps one string to another
- Member functions are provided to search and retrieve items, find items, etc
- The [ ] is overloaded: myMap[key] gives the value from the map associated with the key
- Note: using operator[] creates an entry if one does not exist
- Note: to remove an entry you need to use the erase() method

## Searching

- Because operator [] creates an entry you cannot say something like

```
   if( myMap[key] == “” )
```

- That would create an entry if one did not exist (and it would always be true!)

- Use find() and compare the result to end()

```
if( myMap.find(key) == myMap.end() )
   //key not in the map
```

## Example map use

```
map<string,int> counters;
string word;

while( cin >> word )
      counters[word]++;
      // if counters[…] doesn’t exist, it gets created!

// at the end of this loop, counters contains an entry
// for each word in input.
// The key is the word.
// The value is count of # of times the word appeared
```

## Looking at the entire map

```
map<string,int>::iterator it;

for( it = counters.begin();
             it != counters.end(); it++ )
      cout << it->first << ":" << it->second << endl;
```

NJIT New Jersey's Science & Technology University COLLEGE OF COMPUTING SCIENCES

## Another mechanism

```
map<int,string> myMap;
// …

for( const pair<int,string>& i : myMap ){
   cout << i.first << ":"
            << i.second << endl;
}

// note the map manages a pair<> of items
```

NJIT New Jersey's Science & Technology University COLLEGE OF COMPUTING SCIENCES

## A word about map iterators

- The iterator cycles through the map in “key order”
- So if I want something sorted by key… I can put things into the map and just use an iterator… presto, sorted by key!

## More iterating

- Iterating goes in sorted order; you can go in reverse order by swapping the sense of end and begin, decrementing instead of incrementing, and playing with the edge cases
- OR you can use a reverse iterator

## Same result…

```
for(map<int,int>::iterator it = xx.end();
     it-- != xx.begin(); /* */ )
       cout << it->first << ":" << it->second << endl;

for(map<int,int>::reverse_iterator it = xx.rbegin();
     it != xx.rend(); it++)
       cout << it->first << ":" << it->second << endl;
```

NJIT New Jersey's Science & Technology University COLLEGE OF COMPUTING SCIENCES

## Use maps to sort maps

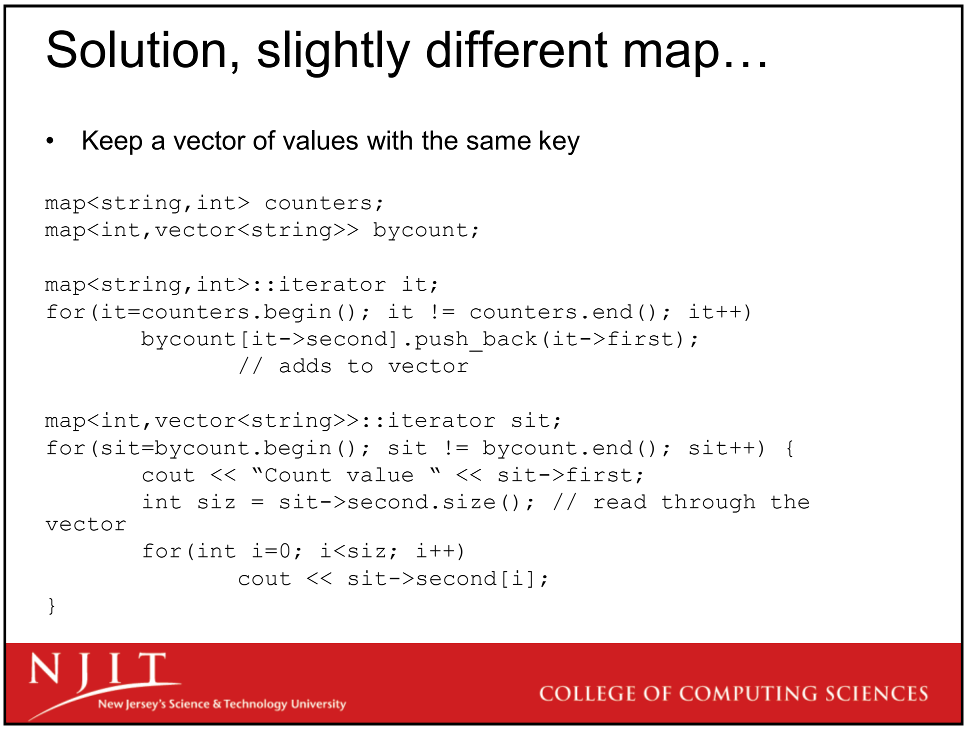
- If I have a map<A,B> and I want to sort it by B, can I just create a map<B,A> and iterate over it?

```
map<string,int>counters;
map<int,string>bycount;

map<string,int>::iterator it;
for(it = counters.begin(); it != counters.end(); it++ )
      bycount[ it->second ] = it->first;

// Problem, though… the keys must be unique…
what if there are two keys in "counters" with
the same value??
```

NJIT New Jersey's Science & Technology University COLLEGE OF COMPUTING SCIENCES



## Solution, slightly different map…

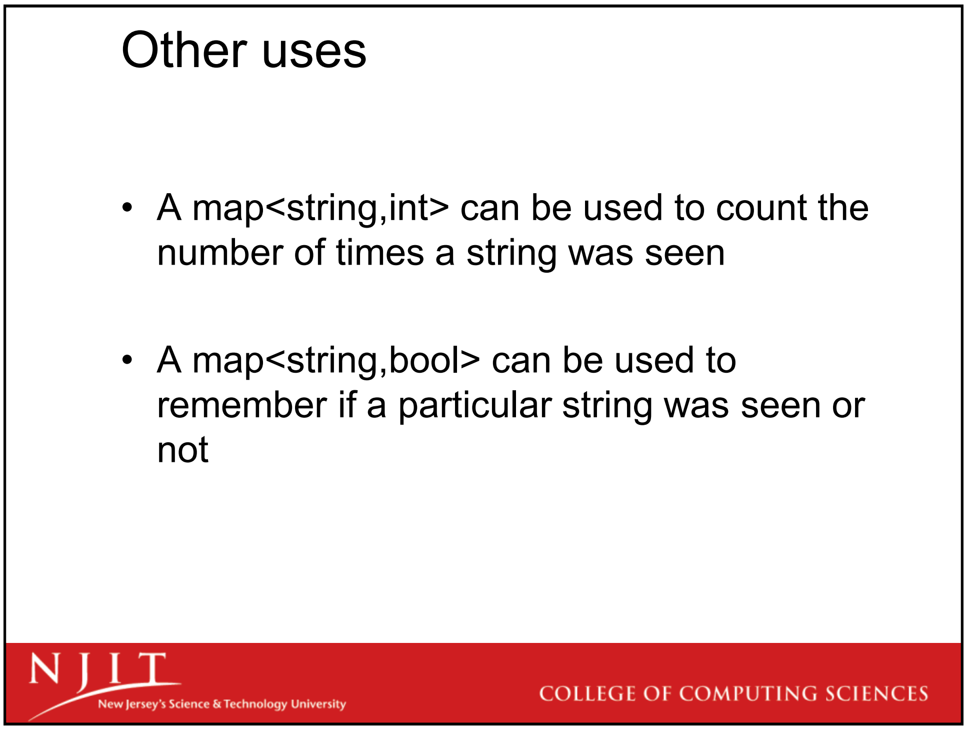
- Keep a vector of values with the same key

```
map<string,int> counters;
map<int,vector<string>> bycount;

map<string,int>::iterator it;
for(it=counters.begin(); it != counters.end(); it++)
       bycount[it->second].push_back(it->first);
              // adds to vector

map<int,vector<string>>::iterator sit;
for(sit=bycount.begin(); sit != bycount.end(); sit++) {
       cout << "Count value " << sit->first;
       int siz = sit->second.size(); // read through the
vector
       for(int i=0; i<siz; i++)
              cout << sit->second[i];
}
```

NJIT New Jersey's Science & Technology University COLLEGE OF COMPUTING SCIENCES



## Other uses

- A map<string,int> can be used to count the number of times a string was seen
- A map<string,bool> can be used to remember if a particular string was seen or not

NJIT New Jersey's Science & Technology University COLLEGE OF COMPUTING SCIENCES

NJIT

THE EDGE IN KNOWLEDGE